

Computational framework for generating large panoramic super-resolution images from localization microscopy

YUE DU,¹ CHENZE WANG,² CHEN ZHANG,¹ LINGYUN GUO,²
YANZHU CHEN,² MENG YAN,² QIANGHUI FENG,² MINGTAO SHANG,¹
WEIBING KUANG,¹ ZHENGXIA WANG,^{2,4} AND ZHEN-LI HUANG^{3,5} 

¹*Britton Chance Center and MoE Key Laboratory for Biomedical Photonics, School of Engineering Sciences, Wuhan National Laboratory for Optoelectronics-Huazhong University of Science and Technology, Wuhan 430074, China*

²*School of Computer Science and Technology, Hainan University, Haikou 570228, China*

³*School of Biomedical Engineering, Hainan University, Haikou 570228, China*

⁴*zxiawang@hainanu.edu.cn*

⁵*Huang2020@hainanu.edu.cn*

Abstract: Combining super-resolution localization microscopy with pathology creates new opportunities for biomedical researches. This combination requires a suitable image mosaic method for generating a panoramic image from many overlapping super-resolution images. However, current image mosaic methods are not suitable for this purpose. Here we proposed a computational framework and developed an image mosaic method called NanoStitcher. We generated ground truth datasets and defined criteria to evaluate this computational framework. We used both simulated and experimental datasets to prove that NanoStitcher exhibits better performance than two representative image mosaic methods. This study is helpful for the mature of super-resolution digital pathology.

© 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Super-resolution localization microscopy (SRLM) is a powerful tool in various biomedical researches by providing great opportunities in resolving biological structures at the nanoscale [1–3]. Recently, researchers have been trying to combine SRLM with pathology to bring valuable molecular insights for the diagnostics and treatment of diseases. For example, Creech et al used SRLM to uncover the ultrastructural distributions of several important proteins (TOM2, Lamin B1 and HER2) in human breast cancer specimens, and thus offer important extension to electron microscopy-based digital pathology [4]. Lennon and co-workers applied quantitative SRLM to the plasma samples from pancreatic ductal adenocarcinoma patients, and identified characteristic populations of extracellular vesicles enriched by the pancreatic cancer [5]. Therefore, as a new member in the digital pathology family, super-resolution digital pathology is bringing pathology to the next level by creating new opportunities for quantitative molecular diagnostics and early disease detection.

Biologists and clinicians are interested in obtaining a panoramic image of the entire sample under studies, so that they can capture rare events or enable a better statistics [6]. Note that the probability of discovering rare events increases with expanding the spatial extent of the image. However, because the size of a microscopic image under a single acquisition is often smaller than the size of biological specimens (especially clinical samples), it is necessary to use an image mosaic method to generate a panoramic image, and a large amount of super-resolution (SR) images from overlapping field of view (FOV) should be carefully stitched. However, these SR images usually result in operation of large files that may be beyond the capability of existing image mosaic methods.

The reported image mosaic methods have been mostly designed for grayscale images from photography or high-resolution microscopy, and usually require information on the sequence of images to be stitched. For example, the MIST plugin can realize automatic mosaicking if the sequence of the microscopy images is obtained [7]. TrakEM2 can reconstruct neural circuit from serial section electron microscopy images [8], but needs to collect the initial stitching positions of the images. Recently, the BigStitcher plugin was developed to provide an interactive stepwise stitching for light-sheet microscopy images [9]. However, when these methods are applied to a large amount of SR images, we face various difficulties, for example, frequent manual input, poor mosaic images, or even software crash due to large image size. Therefore, currently there are no effective methods for automatically generating a large panoramic SR image from a large amount of SRLM images. On the other hand, if we want to use an automatic x-y translational stage to record the absolute position of each sub-image for further image stitching, the stage should have centimeter-scale travel distance and nanometer-scale repeatability, and the latter should be better than the typical resolution of 30–50 nm in SRLM. Unfortunately, such an automatic x-y translational stage is hard to work with inverted fluorescence microscopy, and may be too expensive (USD 50–60k) for some biomedical labs.

Before trying to develop a new image mosaic method, we analyzed the image reconstruction process in SRLM. We noticed that, in SRLM, a localization table is firstly obtained through applying a suitable molecule localization algorithm to the raw images, and then a grayscale image is rendered using information from this table. However, for classical optical microscopy or other super-resolution microscopy techniques, grayscale images are directly generated during the imaging process. It is interesting to find that, although a localization table contains all information for the following image reconstruction, the file size for this localization table is usually smaller than that of the corresponding SR image. For example, if we want to obtain a panoramic image with a FOV of $1\text{ mm} \times 1\text{ mm}$, we need to use 10×10 SRLM images with a FOV of $100\text{ }\mu\text{m} \times 100\text{ }\mu\text{m}$ in each image. When we render the grayscale images with a pixel size of 10 nm, the final file size of the SR image is 40 GB, while the file size of the corresponding localization table is only 5.4 GB. And, if necessary, we can use the same localization table to render a SR image with a smaller pixel size (thus better resolution), without requiring more data storage resource. That is to say, using localization table instead of grayscale images in image mosaicking may bring new chances to solve the challenges in generating large panoramic SR images from SRLM.

In this paper we proposed a computational framework for generating large panoramic SR images from localization table. The framework includes five steps and all steps can be performed automatically. We generated ground truth datasets and defined criteria to evaluate each step in this computational framework. Basing on this computational framework, we developed an image mosaic method called NanoStitcher, for the purpose of stitching SR images. We used both simulated and experimental datasets to compare the performance of NanoStitcher with two representative image mosaic methods (TrakEM2 [8] and BigStitcher [9]) in SR image stitching, and found out that NanoStitcher exhibits the best performance. We believe this study makes a significant step for the mature of super-resolution digital pathology.

2. Methods

2.1. Generating experimental dataset

Experimental datasets were used to evaluate the performance of our proposed computational framework. A complete procedure for generating experimental super-resolution images includes cell culture, sample preparation, imaging and data processing. Below are brief descriptions for each step.

2.1.1. Cell culture

U2OS cells were cultured at 37°C with 5% CO₂ and McCoy 5A medium containing FBS (10%), penicillin (100 U/ml) and streptomycin (0.1 mg/ml). Cells were seeded on a glass bottom dish (MatTek, P35G-1.5-14-C) and grown for 24 hours before fixation.

2.1.2. Sample preparation

Before fixation, cells were rinsed once with pre-warmed PBS. Cells were fixed with fixative solution of 3% paraformaldehyde (Electron microscopy sciences, 157–8), 0.1% glutaraldehyde (Electron microscopy science, 16020) and 0.2% Triton X-100 (Sigma, X100-100ML), soaked in PBS for 15 minutes, and rinsed three times in PBS for 10 minutes each. Cells were infiltrated with 0.2% Triton X-100 in PBS, and washed twice in PBS for 5 minutes each. Next, cells were placed in blocking buffer (3% BSA in PBS (Jackson ImmunoResearch, AB_2336946)) for 1 hour, and incubated with anti- α -tubulin (Sigma, T5168, 1:400 dilution) in blocking buffer for 1 hour. Cells were washed three times in PBS for 10 minutes each, incubated with Alexa Fluor 647 labelled secondary IgG antibody solution (Invitrogen, A-21235, 1:400 dilution) in blocking buffer for 40 minutes, and finally washed three times in PBS for 10 minutes each.

2.1.3. Imaging and data processing

Cells were imaged on a home-built super-resolution localization microscopy system, which was mainly consisted of an Olympus IX73 microscope, an Olympus 60 \times /NA1.42 oil-immersion objective, a 405 nm activation laser, a 640 nm excitation laser, and a Hamamatsu Flash 4.0 V3 camera with an exposure time of 10 millisecond. The lasers were both from LaserWave, China. The FOV was 100 μ m \times 100 μ m. The focal plane in each FOV was roughly determined through an active focus stabilization method, which uses the reflection of 850 nm LED from the coverslip to measure the relative focal position. Single molecule imaging at different z positions were taken to find a position with best signal-to-noise ratio. The intensity of the 405 nm laser was adjusted to ensure a predetermined molecule density during the whole imaging process. The focus was stabilized every two hundred frames.

A two-dimensional automatic microscope stage (XWJ-150R-2, Sanying MotionControl Instruments Ltd, China) was used to move the sample across a large area, with each FOV of 100 μ m \times 100 μ m. After automatic stabilization in a new FOV, sample was illuminated with the 640 nm laser (\sim 8 kW/cm²) for about 1 second to bleach off most of the fluorescent molecules. Images of two adjacent FOVs were controlled to have \sim 10% overlap. During the imaging of each FOV, sample drift was minimized using the active NIR stabilization method.

2.2. Generating ground truth dataset

To verify the feasibility of the proposed computational framework, we selected a SR image of microtubules in U2OS cells to construct a simulation dataset with ground truth (Fig. 1(a)). The SR image has localization points evenly distributed throughout the whole imaging area (or region of interest, ROI), so that after splitting the original SR image we can obtain a set of successive images with sufficient microtubule structures across each image. The original SR image consists of 952 \times 944 pixels, and each pixel corresponds to 108 nm in sample plane. Originally the localization table for this SR image has 28,239,203 localization points. After removing and adding noise, the number becomes 27,766,927. The size of the localization table is 1.2 GB. We divided the SR image into 5 \times 5 sub-images (see Fig. 1(b)). The width of the overlapping region between each two adjacent sub-images is 6 pixel.

The generation process of 5 \times 5 sub-images, or localization tables, was divided into three steps: sparsification, noise addition, and cutting (see Fig. 1(c)). Briefly, we identified the overlapping regions with a width of 6 pixels, sparsified and added noise to the corresponding localization table of the overlapping regions, then cut the whole image to a set of sub-images.

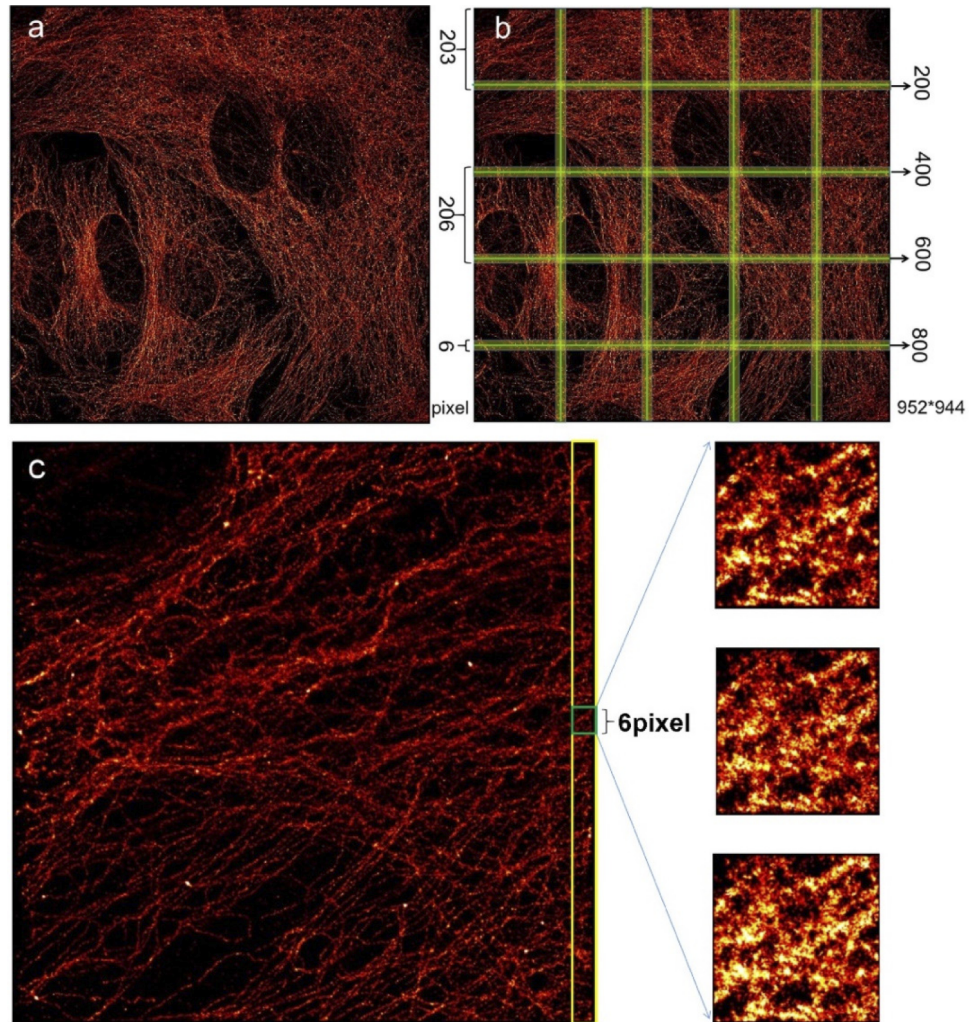


Fig. 1. Generating ground truth datasets. (a) The ground truth image. (b) The locations for cutting (the green lines) and the overlapping regions (the green strips). (c) A representative sub-image. The left image was cut from the upper-left corner in (b). The three small ROIs in the right are presented to illustrate the processing of overlapping regions. Top: raw image; Middle: sparsified image after removing 20% localization points; Bottom: image with 1% noise point.

2.2.1. Sparsification

The signal intensities at the boundary pixels of an experimental SR image are normally weaker than those in the center of the image. To account for this situation, voxel grid filtering method was used to realize a signal gradient in the boundary pixels. Specifically, the overlapping regions with a width of 6 pixel were sparsified by voxel grid filtering, where the size of the grid was 0.07 and 15–20% of the localization points were removed.

2.2.2. Noise addition

In the vertical overlapping regions in Fig. 1(b), i.e. $x \in (197, 203), (397, 403), (597, 603), (797, 803)$, the number of localization points is 611720, 651399, 675560, 669387, respectively. A total of 8000 random noise points were added to each of the above overlapping regions, accounting for about 1% of the total points in each overlapping region. The x and y coordinates of these random noise points were generated randomly within the coordinate range of the overlapping region. The other localization information for the noise points came from randomly selected localization points in the unprocessed localization table, and all of their localization information except for the coordinates was copied. The addition of noise points in the horizontal overlapping region was the same as above.

2.2.3. Cutting

The whole SR image was cut into 5×5 sub-images. The overlapping region in each adjacent sub-image is 6 pixel.

2.3. Computational framework for generating large panoramic super-resolution images

The proposed computational framework for stitching multiple SR images includes five steps, including overlapping region selection, point set data preprocessing, pairwise image stitching, stitching path optimization, and patchwork fusion and smoothing (as show in Fig. 2). Briefly, the overlapping region between two stitched images is firstly calculated using cosine similarity, and the matching region is determined automatically. Then, to enhance the performance of the following matching process, statistical filtering method is used to reprocess the overlapping regions. Next, after pairwise image stitching, path optimization is applied to find a path with minimum error for stitching all SR images. Lastly, the patchwork is fused and smoothed to obtain a final panoramic image. This computational framework was designed to enable a method called NanoStitcher, for the purpose of stitching images at nanometer scale resolution.

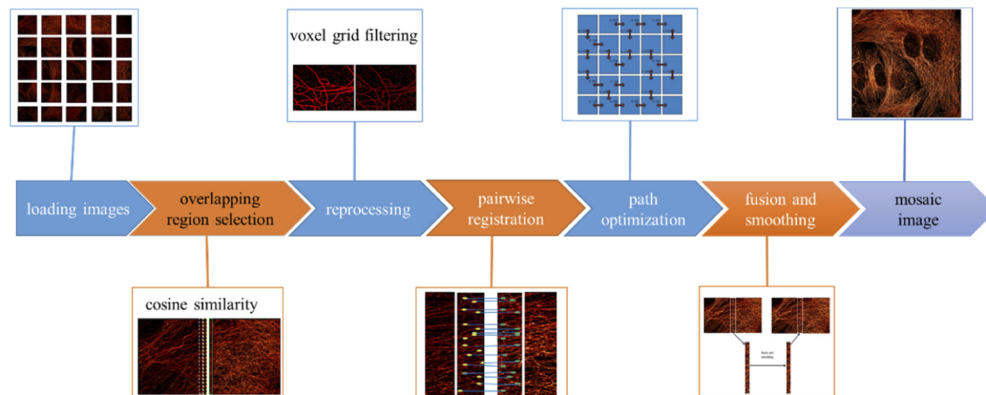


Fig. 2. The computational framework for generating a panoramic super-resolution image.

2.3.1. Overlapping region selection

To stitch a large panoramic image effectively, the overlapping regions between adjacent images should be determined. In one of the two images that needs to be stitched, a region with an appropriate width (determined by the experimental SR images) from the edge of the image is selected (see the green dashed box in Fig. 3(b)). Then, a region with the same size is slid gradually in the direction of the red arrow across the other image to be stitched, from the edge to the inner of the image (Fig. 3(a)). To find the most similar overlapping region between these two images, cosine similarity is computed for each pair of slides. Cosine similarity [10] evaluates the similarity of two vectors by calculating the cosine of the angle between them, and the formula is as follows.

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{||A|| ||B||} \quad (1)$$

where A and B is vectors, θ is the angle of A and B . The cosine similarity ranges from -1 to 1 . Here -1 means the two vectors are pointing in exactly opposite directions, and 1 means the two vectors are pointing in the same direction, 0 usually means the two vectors are independent of each other, and a value in between indicates the intermediate similarity and dissimilarity.

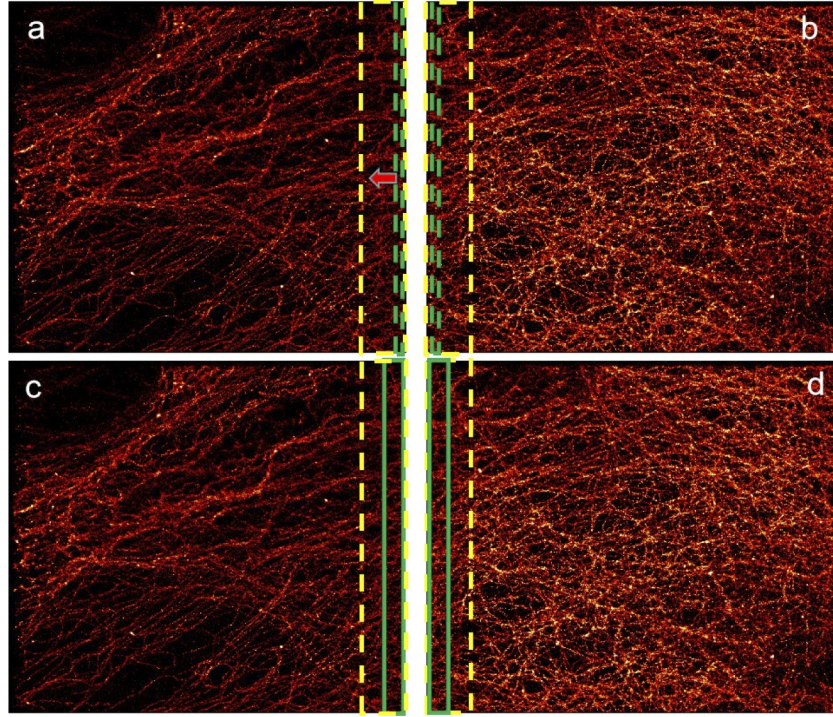


Fig. 3. Illustration on overlapping region selection. (a) One of the adjacent images. The red arrow indicates the direction for searching overlapping region. (b) The other of the adjacent images. (c-d) The adjacent images after overlapping region selection. Here the green dash box represents the pre-determined region used for searching the overlapping region, the green solid box represents the overlapping region, and the yellow dashed box represents the sliding candidate region.

Selection of the width of the slider is important. If it is too long and/or exceeds the experimental overlapping region, the similarity is poor and the appropriate overlapping region cannot be found. If the width is too short, the calculation time will be too long, and the time complexity is

increased. In this paper, each experimental SR image to be stitched is 1040×1040 pixels (which is different from the SR image used for generating simulation ground truth dataset) and the experimental overlapping ratio is about 5%, meaning that the width of experimental overlapping region is about 50 pixels. Therefore, the width of slider is set to 20 pixels. Vertical alignment works the same way. The final overlapping region is shown in Fig. (3c)-(3d).

2.3.2. Point set data preprocessing

Original SR images are reconstructed from a huge amount of localization points from both signal and background noise, which will affect the performance and the speed of the stitching algorithm. Background noise can be reduced by simply averaging pixels with uneven brightness. However, because the signal at the image boundary is relatively weak, many pixels are considered as noise and will be removed by this averaging process. This treatment could damage the biological structures in the image, or result in structure disconnection. Therefore, a desirable method for data reprocessing should reduce the background noise and the amount of the localization points, without destroying the connection of biological structures (such as microtubules) in the image. Here we use a voxel grid filtering method to preprocessing the data. The benefits are, in one side, removing noise and outliers can improve the accuracy of the registration algorithm; on the other side, reducing the amount of localization points (and thus the data size) can accelerate the registration speed.

In each voxel grid, we use the average coordinate value of all localization point to represent all the localization points of the entire grid. Since the data preprocessing here reduces noise and some points only to improve the accuracy and speed of the calculation of offset in pairwise registration, all localization points are still joined together after the calculation of registration offset, so only the coordinate average values of the registration point can be considered here. Figure 4 shows the dot plot before and after voxel grid filtering preprocessing, and it can be clearly observed that the processed image reduces noise while maintaining the connectivity of the microtubule structure. Note filtered images are merely used to calculate XY coordinate offsets between pairwise images, and a final panoramic super-resolution image is generated using coordinate offsets and original super-resolution images, rather than filtered images.

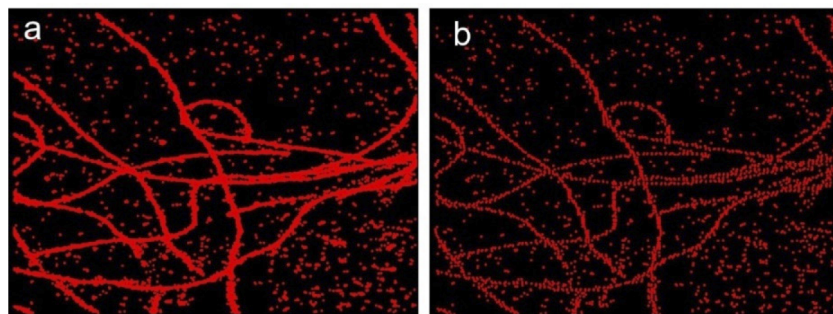


Fig. 4. Dot plot before and after voxel grid filtering. (a) Before the filtering. (b) After the filtering.

2.3.3. Pairwise image stitching

After considering that each localization table contains a large number of localization points, we select the reported joint registration of multiple point clouds (JRMPC) algorithm [11] to perform the pairwise image stitching for both the experimental and simulated localization datasets. We compare the JRMPC algorithm with two classical pairwise point set registration algorithms,

including Iterative Closest Point (ICP) [12] and Singular Value Decomposition (SVD) [13]. Here we briefly introduce the principle of using JRMP algorithm in pairwise image registration.

The basic idea is to transform the coordinates of two given point sets V_1 and V_2 to the same spatial coordinate system, and thus form a new point set Θ .

$$\begin{aligned} V_{1_tran} &= R_1 * V_1 + t_1, \\ V_{2_tran} &= R_2 * V_2 + t_2, \\ \Theta &= \{V_{1_tran}, V_{2_tran}\}. \end{aligned} \quad (2)$$

To find the transformation parameters (R_1, R_2, t_1, t_2) with the correct matching between point set V_1 and V_2 , the steps are as follows.

Step 1: Randomly initialize a series of points, and set them as grid points in the DATA size dimension. The K -dimensional Gaussian mixture model (GMM) is built with these points as the centers, where K is the number of initialized points. Initial values should be given as the parameters in the GMM, and will be updated alliteratively later.

Step 2: Gaussian model is used to calculate the values of each point x in the point set $\Theta(i)$, which is divided into the K th Gaussian model.

$$p(x) = \sum_{k=1}^K \pi_k N(\Theta(x) | \mu_k, \Sigma_k) \quad (3)$$

where π_k are the mixing coefficients that satisfy $\sum_{k=1}^K \pi_k = 1$, while μ_k and Σ_k are the mean and covariance matrices.

Step 3: Max $f()$ was solved by expectation-maximization algorithm (EM), and it is known that the maximum $f()$ can be obtained when the i th point is assigned to be the K th Gaussian model. Thus the classification results are obtained.

$$f(\Theta) = \sum_{ik} p(x) \quad (4)$$

Step 4: Calculate the transformation parameters (R_1, R_2, t_1, t_2) of the point set, the detail can be found in literature [11].

Step 5: Based on the point set generated by the new transformation parameters and the GMM, EM algorithm is used to solve the problem again. Iterate the whole process until the specified number of iterations is met.

2.3.4. Stitching path optimization

A panoramic image is obtained by cumulative stitching of sub-images. In this process, there are different stitching paths to choose. Note that the registration performance is affected by noise, the quality of overlapping regions and many other factors. Therefore, the registration quality between different adjacent sub-images will accumulate into a big difference in the quality of a final panoramic image. Clearly, it is important to choose an optimal stitching path. Through experimental verification, we found that the Euclidean distance can quantify the degree of overlap for two adjacent regions after registration. Here we take the Euclidean distance as the evaluation index for stitching path optimization, and use the minimum spanning tree to find the best path.

The optimization of stitching path is as follows. Firstly, we select and regard a sub-image as the vertex, and calculate the Euclidean distances between each pair of sub-images. Then, we regard these distances as the weight of the edge, and obtain a weighted graph. Finally, we use the minimum spanning tree algorithm to obtain the shortest path of the graph. In this way, we obtain an optimal path for image stitching.

2.3.5. Patchwork fusion and smoothing

After completing the point set stitching, the datasets from two adjacent localization tables are transformed into the same coordinate system, according to the coordinate offset calculated in the *pairwise image alignment* step. Therefore, the number of localization points in the patchwork (or overlapping regions) would be doubled as compared with the actual localization points. Patchwork fusion is a process for blending the registration results into a uniform point set according to a specific fusion method. However, the localization points obtained from experiments contain a lot of information, including but not limited to signal intensity, and localization accuracy in different dimensions. It is difficult to estimate these information for a new localization point using the information from adjacent localization points. Therefore, we only remove duplicate points during the fusion process, so that the original data quality is not affected.

We adopt voxel grid filter method to remove the duplicate points. Briefly, we divide an overlapping region into many small voxel grids. In each voxel grid, we use a random localization point to represent all the localization points in the grid, and thus effectively remove the duplicates. The steps are as follows: 1) Establish the adjacency matrix according to the adjacency relation between two localization tables, and find two localization tables that are adjacent to each other. 2) Divide the overlapping region in the two adjacent localization tables into many small voxel grids. 3) Replace the localization points in the whole overlapping region with randomly selected localization points in the small grids.

2.4. Quantifying the performance of the computational framework

We choose three parameters to evaluate the performance of the proposed computation framework for SR image stitching. These parameters are from representative metrics for comparing the quality of point set or grayscale image.

2.4.1. Structural similarity index

Structural similarity index (SSIM) is a measure of structural similarity between two grayscale images [14]. SSIM is calculated from a combination of three different factors, including brightness, contrast, and structure similarity between samples. The mean value is used to estimate brightness, the standard deviation to estimate contrast, and the covariance to measure structural similarity. The equation for calculating SSIM is:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (5)$$

where μ_x, μ_y is the mean value of x and y , σ_x^2, σ_y^2 is the standard deviation of x and y , and σ_{xy} is the covariance. Two constants, $c_1 = (k_1L)^2, c_2 = (k_2L)^2$, are used to maintain stability, L is the dynamic range of pixel values, k_1 is set to be 0.01, and k_2 is set to be 0.03.

SSIM ranges from -1 to 1 . When two images are identical, SSIM is equal to 1 . We use SSIM to quantify the overlapping regions obtained from an image stitching method with the corresponding regions from ground truth.

2.4.2. Mean square error

Mean Square Error (MSE) measures the average squared difference between the estimated values and the true value of a parameter [15]. In image processing, MSE is the mean squared sum of the pixel value difference between a processed image and the original image.

$$MSE = \frac{1}{M * N} \sum_{i=1}^M \sum_{j=1}^N (f'(i, j) - f(i, j))^2 \quad (6)$$

where $f(i,j)$ is the original image, and $f'(i,j)$ is the image to be evaluated. M and N represent the length and width of the image, respectively. Note that a smaller MSE value means a better similarity in the two images.

2.4.3. Chamfer distance

Chamfer Distance (CD) is simple and effective metric for comparing point sets [16]. CD provides high-quality comparison on two non-correlating point sets, without requiring the same number of points in these point sets. We calculate CD between a transformed source point set S_1 and a target point set S_2 .

$$d_{CD}(S_1, S_2) = \frac{1}{S_1} \sum_{x \in S_1} \min_{y \in S_2} \|x - y\|_2^2 + \frac{1}{S_2} \sum_{y \in S_2} \min_{x \in S_1} \|y - x\|_2^2 \quad (7)$$

where S_1 and S_2 represent two point sets. The first term is the sum of the minimum distance from any point x in S_1 to S_2 , and the second term is the sum of the minimum distance from any point y in S_2 to S_1 . A smaller CD means a better match in two point sets.

The image stitching in this paper was based on point sets (localization table), but it is difficult to evaluate the image stitching quality directly from the point sets themselves. Therefore, a pair of point sets were extracted from the overlapping region in two adjacent sub-images, and the stitching quality was calculated by evaluating the similarity of these two point sets. We used CD to quantify the similarity of two point sets. For simplicity, in this paper, CD was calculated using a pair of point sets with the same number of localization points.

3. Results

3.1. Description of the experimental and ground truth datasets

In this paper, we verified the performance of our proposed method on two kinds of datasets: experimental datasets without ground truth, and simulated datasets with ground truth. The experimental datasets were from SR imaging of microtubules, and the datasets consist of 25 ROIs (or localization tables). The field of view in each ROI is $100 \mu\text{m} \times 100 \mu\text{m}$. In these ROIs, the average number of localization points is 3,017,130, and the average storage size is 138.96 M. Table 1 shows information from a representative ROI, where the number of localization points is close to the average value. The size of the rendered image increases significantly from 4.12 MB (100 nm rendering) to 1610 MB (5 nm rendering). Note that the pixel size for rendering should be 10 nm or smaller, if we don't want to lose the image quality from the localization tables. The SR image rendered at 10 nm for this ROI is shown in Fig. 5(a). The zoom-in images rendered at 10 nm and 5 nm for this ROI is shown in Fig. 5(b) and Fig. 5(c), respectively.

Table 1. Information of Localization Table and the Rendered SR Images in Experimental and Simulated Datasets

Dataset	The number of points	Size of localization table (MB)	Size of images rendered with different pixel sizes (MB)							
			100 nm		50 nm		10 nm		5 nm	
			PNG	TIF	PNG	TIF	PNG	TIF	PNG	TIF
Experimental data	2922882	133	0.28	4.12	0.85	16.5	12	412	28.5	1610
Simulated data	1283934	58.7	0.03	0.17	0.12	0.7	1.88	17.1	5.06	73.8

The simulated datasets were also SR imaging of microtubule structures, but the datasets consist of only one ROI with 952×944 pixels, and each pixel corresponds to 108 nm in the sample plane. During the experiments, we captured as many localization points as possible to obtain a high quality SR image. Finally, we had a total of 27,766,927 localization points in the ROI. As shown in Fig. 1, we divided the reconstructed SR image from this ROI into 5×5 sub-images,

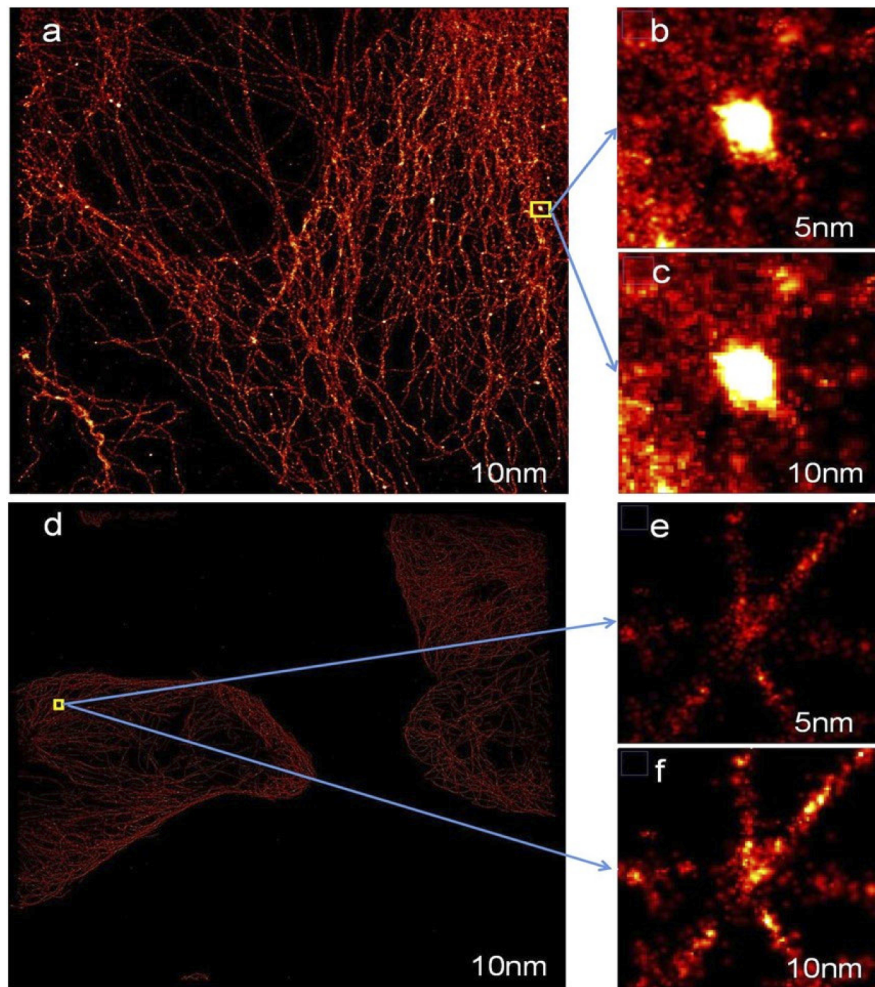


Fig. 5. Super-resolution images of the selected ROIs presented in Table 1. (a) Simulated dataset. (b,e) Re-rendered image for the yellow box in (a) or (d), using a pixel size of 5 nm. (c,f) Re-rendered image for the yellow box in (a) or (d), using a pixel size of 10 nm. (d) Experimental dataset.

corresponding to 25 sub-localization tables. The average number of localization points in each sub-image is 1,255,645, and the average storage size of a sub-image is 57.4 MB. Table 1 shows information from a representative sub-image where the number of localization points is close to the average value. The storage size of localization tables and the corresponding greyscale images rendered with different pixel sizes are also shown in Table 1. The SR image rendered at 10 nm for this ROI is shown in Fig. 5(d). The zoom-in images rendered at 10 nm and 5 nm for this ROI is shown in Fig. 5(e) and Fig. 5(f), respectively.

It is worthy to note that, for each localization point in experimental or simulated datasets, a complete localization table would contain 12 parameters, including x coordinate, y coordinate, z coordinate, peak intensity (photon), PSFSigmaY (pixel), PSFSigmaX (pixel), total intensity (photon), background (photon), SNR (peak to background, e-), CRLBx (nm), CRLBy (nm), and frame number. Because each parameter is float type and occupies 4 Bytes, each localization point requires a storage size of 48 Bytes. If we choose to save a minimal number of parameters

required for SR image rendering, which includes x coordinate, y coordinate, total intensity (photon), CRLB (nm) and frame number, the storage size for each localization point could be further reduced to 20 Bytes.

3.2. Performance quantification on simulated datasets

The mosaic process of panoramic super-resolution image consists of the five steps mentioned above. First, we used the cosine similarity to compute the overlapping region of two adjacent localization table, the width of the overlapping region is about 6 pixels. In the process of simulating data generation, we set the overlapping region of all adjacent localization tables to be the same value. Secondly, we used the voxel grid filtering method with the grid size of 0.5 to reprocess the localization table, and complete the subsampling of the localization table. Thirdly, three point set registration methods were used to complete pairwise stitching of super-resolution images. Then, the stitch path was optimized using minimum spanning tree. Finally, fusion and smoothing were done using pixel grid filter method with the grid size of 0.09 and 0.01 respectively.

3.2.1. Pairwise image stitching

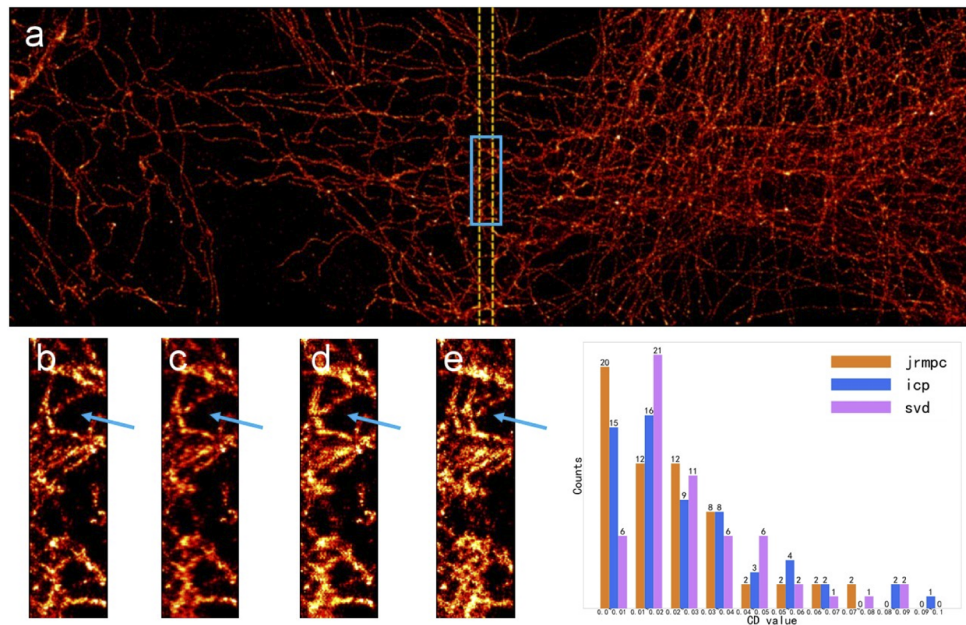


Fig. 6. Horizontal pairwise stitching results from different algorithms. (a) Ground truth of stitched super-resolution image. (b) Ground truth of the blue box in (a). Enlarged stitched images from JRMPCC (c), ICP (d), SVD (e), which are from the blue box in (a). (f) CD histogram calculated from the two overlapping regions shown by the yellow box in (a). The arrows in (b-e) indicates a representative structure for comparison.

We evaluated the performance of three registration algorithms (JRMPC, ICP and SVD) in pairwise point set alignment of super-resolution localization microscopy images. We selected two adjacent localization tables in the horizontal and vertical directions from the simulated datasets, and stitched them with the above algorithms. We showed the results after rendering in Fig. 6, 7, and used three different indexes (CD, SSIM, and MSE) to quantify the alignment results.

We took a close look at the images in Fig. 6.7, and found out that the stitched images made by JRMPC and ICP algorithms have a good visual impression, without obvious stitching dislocation

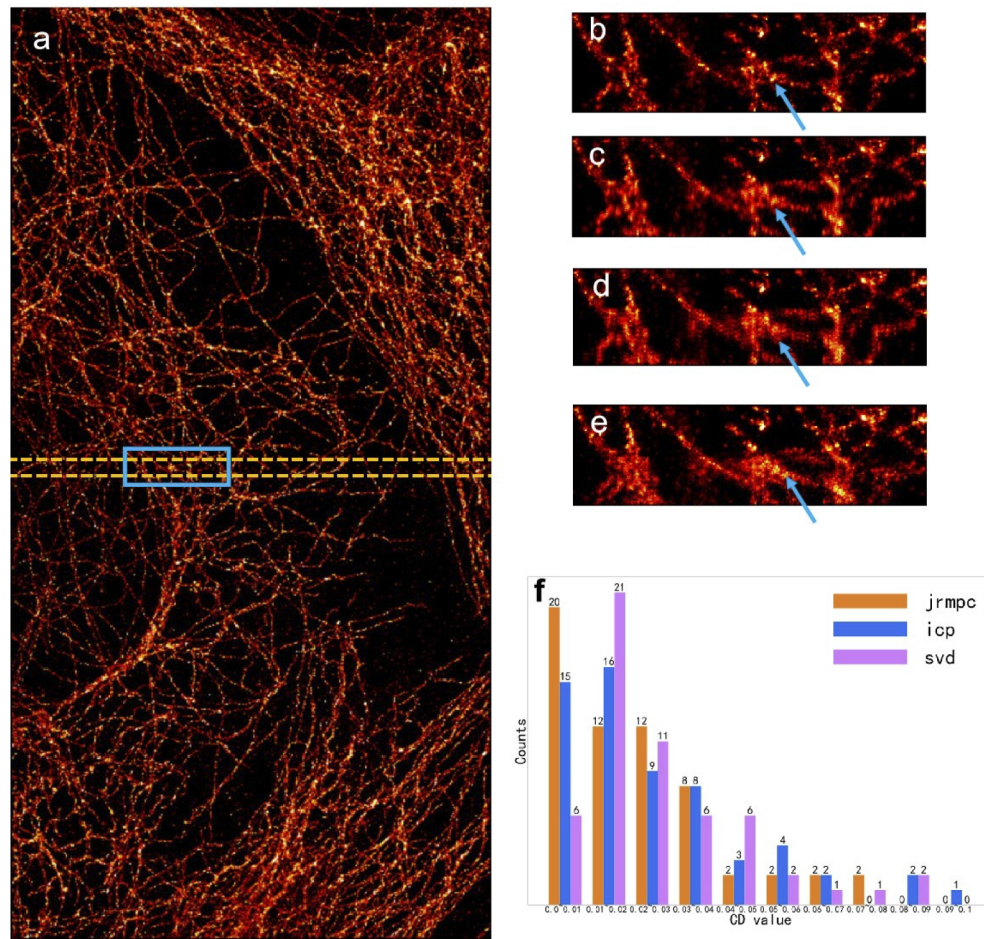


Fig. 7. Vertical pairwise stitching results from different algorithms. (a) Ground truth of stitched super-resolution image. (b) Ground truth of the blue box in (a). Enlarged stitched images from JRMPC (c), ICP (d), SVD (e), which are from the blue box in (a). (f) CD histogram calculated from the two overlapping regions shown by the green box in (a). The arrows in (b-e) indicates a representative structure for comparison.

or broken filamentous structures. However, for the stitched image by SVD algorithm, we observed mismatched structures obvious patchwork joints, as shown by the blue arrows in Fig. 6(e) and Fig. 7(e).

We randomly selected 80 rectangular sub-regions at the overlapping region of the two adjacent localization tables, and calculated chamfering distance (CD) for each pair of these sub-regions after removing the non-evaluable and noise points. We show the distribution histogram of CD values in Fig. 6(f) and Fig. 7(f), and observed that: (1) most of the CD values are distributed between 0 and 0.04, (2) the JRMPC algorithm presents the best CD distribution (more results in the low CD value region) among the three algorithms, and (3) the SVD algorithm presents the worst CD distribution.

We calculated the average CD values for the 80 sub-regions mentioned above. The results are shown in Table 2 and Table 3. JRMPC seems to have the best performance in the average CD value, as well as the offset. Further evaluations using SSIM and MSE also confirm that JRMPC is the best among the three stitching algorithms. Note that for a better stitching algorithm, SSIM is

higher and MSE is lower. Therefore, in the subsequent process for generating panoramic images, we just use the JRMPC algorithm for pairwise image stitching.

Table 2. Evaluating the Horizontal Stitching Performance of Three Algorithms Using Simulated Dataset

Algorithm	CD	MSE	SSIM	Offset on the X-axis (real:197)
JRMPC	0.0180	518.87	0.78	196.75
ICP	0.0198	805.22	0.70	196.52
SVD	0.0245	927.25	0.65	

Table 3. Evaluating the Vertical Stitching Performance of Three Algorithms Using Simulated Dataset

Algorithm	C.D.	MSE	SSIM	Offset on the Y-axis (real: 200)
JRMPC	0.0225	532.09	0.84	199.77
ICP	0.0264	737.08	0.79	199.56
SVD	0.0272	951.84	0.75	200.52

3.2.2. Stitching path optimization

A good stitching on two adjacent sub-images is only the first accomplishment for generating a mosaic image, and further image processing steps would be required to generating a final panoramic image. For a panoramic image stitched from a large number of sub-images, the alignment errors in all stitching tasks will be accumulated. Therefore, it is necessary to plan a reasonable stitching path for the panoramic image stitching, so that the sum of alignment errors could be minimized.

We form a graph by regarding the sub-images as the vertices, and the Euclidean distance between two adjacent joints as the weights of the edges (Fig. 8(a)). First, we use the breadth-first algorithm to obtain the stitching path, and the cumulative value of Euclidean distance is 3.537. Then we use the minimum spanning tree algorithm to optimize the stitching path, and the cumulative value of Euclidean distance becomes 3.149. So, we use the Kruskal's algorithm (a minimum spanning tree algorithm) to optimize the stitching path (Fig. 8(b)). After applying the Kruskal's algorithm to this graph, we can obtain an optimized stitching path diagram (Fig. 8(c)).

3.2.3. Patchwork fusion and smoothing

After finding an optimal stitching path, we use JRMPC algorithm again to align a series of paired sub-images that are defined by the optimal stitching path. In this way, we obtained a large localization table and thus reconstruct a mosaic panoramic image, as shown in Fig. 9(a). However, the overlapping regions are relatively bright due to extra localization points for the same structures. We use voxel grid filter method described in Section 2.3.5 to remove the extra localization points, and smooth the whole image. We present a final smoothed panoramic image in Fig. 9(b), where the patchwork is no longer obvious. Here the grid size for removing and smoothing is 0.09 and 0.01 respectively.

3.2.4. Overall performance quantification on simulated datasets

We compared the overall performance of our proposed method (NanoStitcher) with two reported stitching plus-ins, including BigStitcher [9] and TrakEM2 [8] using simulated datasets. Since these reported plugins can only process grayscale images, we first rendered the localization tables into a set of grayscale sub-images, and then use the reported stitching tools as well as our proposed method to stitch them into a panoramic image. The localization table was from

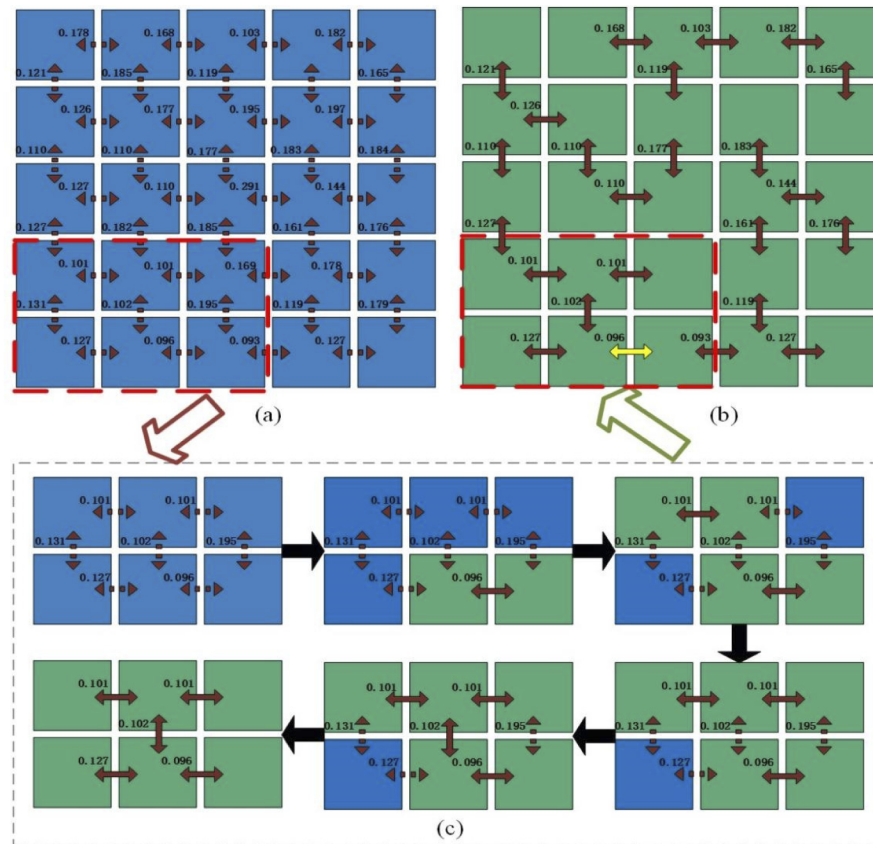


Fig. 8. Stitching path optimization using Kruskal's minimum spanning tree algorithm. (a) Sub-images need to be stitched. Each blue or green square presents a sub-image. The two-way dashed arrow indicates that two adjacent sub-images can be stitched. The numbers next to the arrows are the Euclidean distances calculated from the corresponding points in the two adjacent sub-images. (b) The optimal stitching path after applying the Kruskal's algorithm. The brown two-way solid arrows represent the selected edges, and the yellow two-way solid arrow points out the first selected edge. (c) The optimization process using the Kruskal's algorithm in red dotted box.

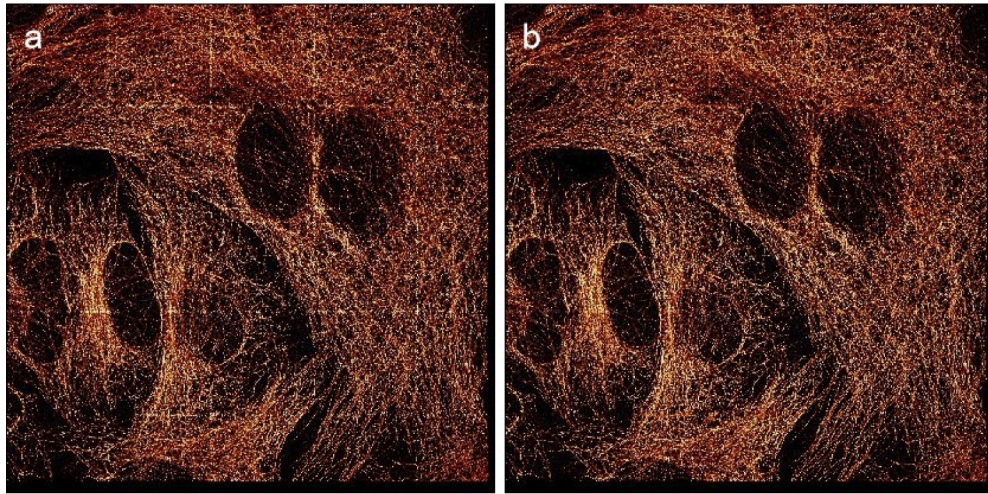


Fig. 9. The mosaic image. (a) Before fusion and smoothing. (b) After fusion and smoothing.

the simulated dataset, and the pixel size for rendering was 10 nm. We generated a total of 25 sub-images, with a total size of 396 MB.

To retain the details of the microtubule structures, we saved the sub-images at TIF format. Rather than BigStitcher and TrakEM2, we also tried several other plus-ins for stitching, including plugins (MIST [7], MasaicJ [17], Grid/collection [18]), but found out that these three stitching plugins cannot process TIF format images. Moreover, TrakEM2 requires the user to specify not only the order of stitching, but also rough positions for each sub-image. Then, TrakEM2 makes fine adjustments to these positions. BigStitcher can automatically complete the stitching of 5×5 sub-images, but the performance is relatively poor, as shown in Fig. 10.

We took a close look at the panoramic images stitched from our method (NanoStitcher) and the reported methods (BigStitcher and TrakEM2). From Fig. 10(b) and Fig. 10(f), we observed that the panoramic image from NanoStitcher presents evenly distributed microtubule structures with no obvious disconnection. However, TrakEM2 presents uneven images and some broken filamentous structures, while BigStitcher provides a panoramic image with poor quality where the details in microtubule structures cannot be observed. Quantitative evaluation of the panoramic images using well established parameters (SSIM and MSE) are also shown in Fig. 10(b-d). Note that these values are obtained by comparing the panoramic images from each method and the Ground truth dataset. NanoStitcher seems to present better results than BigStitcher and TrakEM2.

We also evaluated the performance of NanoStitcher in other types of morphologies, using ground truth datasets generated from SR images of mitochondria in COS-7 cell and membrane proteins (Protein4.1) in red blood cell. The mitochondria structure was not completely sampled, and thus shows a mixture of clusters and short filaments. We verified that NanoStitcher is applicable to these SR images, and that NanoStitcher still provided better performance than BigStitcher and TrakEM2 (data not shown).

3.3. Overall performance quantification on experimental datasets

The experimental dataset consists of 5×5 localization tables, representing a total field of view of $500 \mu\text{m} \times 500 \mu\text{m}$. When the experimental dataset was rendered at 10 nm pixel size, the resulting 25 sub-images had a total size of 10 GB. We were able to use NanoStitcher to complete the stitching, where localization tables rather than sub-images were employed. However, the BigStitcher plugin was unable to stitch these experimental sub-images, and an error was reported

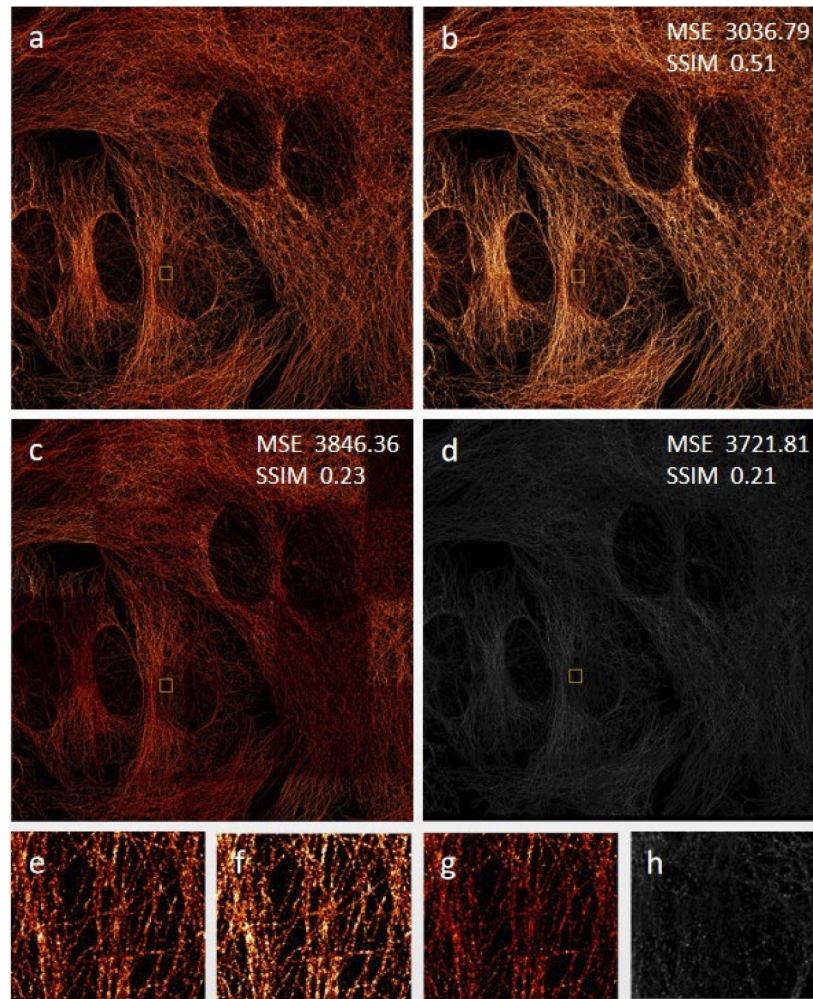


Fig. 10. Panoramic super-resolution images from different stitching methods. (a) Ground truth. (b) NanoStitcher. (c) TrakEM2. (d) BigStitcher. (e-h) Enlarged images for the corresponding yellow boxes in (a-d). A total of 25 simulated datasets were used to generate the images in (a-b).

in the first step of the stitching process. The TrakEM2 plugin can complete the stitching and display a panoramic image. But, after completing the stitching using this plugin, the whole image cannot be saved at full size. The results are shown in Fig. 11, but the panoramic image shown in Fig. 11(b) was actually compressed before stored.

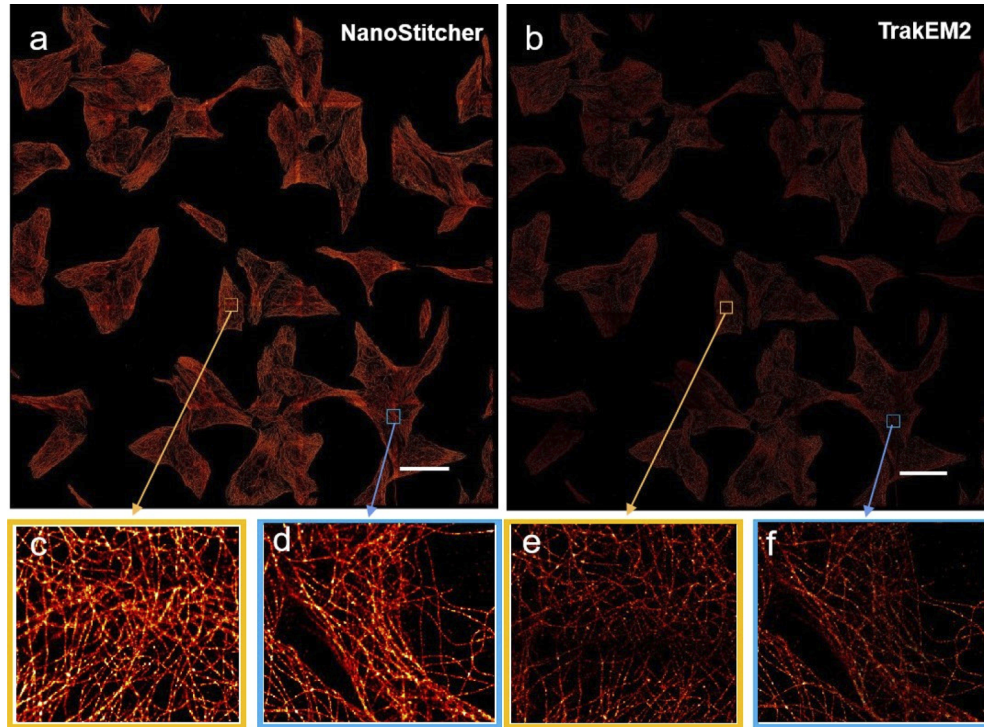


Fig. 11. Panoramic super-resolution images from NanoStitcher and TrakEM2. (a) NanoStitcher. (b) TrakEM2. (c-d) Enlarged image for the yellow and blue boxes in (a). (e-f) Enlarged image for the yellow and blue boxes in (b). A total of 25 experimental datasets were used to generate the images in (a-b). Scale bar: 50 μm (a-b).

Since there is no ground truth in the experimental data, we can only compare the panoramic images using visual observation or line profile calculation. As shown in Fig. 11(a), the panoramic image obtained from our NanoStitcher method presents a satisfactory visual effect, where microtubule structures are clear and continuous and no obvious mismatch can be observed. The zoom-in images in Fig. 11(c-d) confirm these findings. On the other hand, the panoramic image stitched from TrakEM2 is relatively poor for visual observation and presents fuzzy structures. From the zoom-in images (Fig. 11(e-f)), we can clearly observe broken structures, as well as some blurred filamentous threads.

Finally, we are confident to conclude that the computational framework developed in this paper (and the associated NanoStitcher method) is capable of processing experimental datasets with larger data volume, because our computational framework is based on a set of localization tables which have much smaller size than the corresponding sub-images. And, localization tables can retain the original details of the biological structures, and can be used to render a full or panoramic image or a zoom-in image, basing on user-defined parameters or the screen size.

4. Conclusions

In this paper, we proposed a computational framework for generating large panoramic super-resolution images from localization microscopy. The framework consists of five steps, including overlapping region selection, point set data preprocessing, pairwise image stitching, stitching path optimization, and patchwork fusion and smoothing, and all steps can be performed automatically. Unlike traditional image stitching methods based on grayscale images, our framework relies on stitching of localization tables, and thus has advantages on minimized file size, lossless file storage, and adaptive visualization. This computational framework enables a stitching method called NanoStitcher. We compared NanoStitcher with the reported plugins for image stitching (TrakEM2 and BigStitcher), and proved that NanoStitcher provides better overall performance in the image quality.

Instead of using an image stitching method, it may be possible to use nanostructures with well-defined distances (for example, 2D distribution of nanorulers or fiducial markers) to quantify coordinate offsets for further image stitching. However, because the stitching is usually performed in an overlapping area that is from a small portion (10% or less) of a super-resolution image, it is unlikely to have such nanostructures in all overlapping areas, meaning that coordinate offsets from some areas would be missing. That is to say, using nanostructures to register sub-images would face difficulties in generating a panoramic super-resolution image. Nevertheless, this study shows the power of point set-based image stitching methods, and is helpful for the development of super-resolution mosaic microscopy for large biological samples.

Funding. National Natural Science Foundation of China (81827901); Fundamental Research Funds for the Central Universities (2018KFYXKJC039); Natural Science Foundation of Hainan Province (620RC558); Natural Science Foundation Project of CQCSTC (cstc2018jcyjAX0398); Start-up Fund from Hainan University (KYQD(ZR)20022, RZ2100003004).

Acknowledgments. We thank the Optical Bioimaging Core Facility of WNLO-HUST for technical support.

Disclosures. The authors declare that there are no conflicts of interest related to this article.

Data availability. The NanoStitcher method discussed in this paper is not publicly available at this time but will be available shortly in Ref. [19].

References

1. L. Schermelleh, A. Ferrand, T. Huser, C. Eggeling, M. Sauer, O. Biehlmaier, and G. P. C. Drummen, "Super-resolution microscopy demystified," *Nat. Cell Biol.* **21**(1), 72–84 (2019).
2. Y. M. Sigal, R. Zhou, and X. Zhuang, "Visualizing and discovering cellular structures with super-resolution microscopy," *Science* **361**(6405), 880–887 (2018).
3. D. Baddeley and J. Bewersdorf, "Biological insight from super-resolution microscopy: what we can learn from localization-based images," *Annu. Rev. Biochem.* **87**(1), 965–989 (2018).
4. M. K. Creech, J. Wang, X. Nan, and S. L. Gibbs, "Superresolution imaging of clinical formalin fixed paraffin embedded breast cancer with single molecule localization microscopy," *Sci. Rep.* **7**(1), 40766 (2017).
5. K. M. Lennon, D. L. Wakefield, A. L. Maddox, M. S. Brehove, A. N. Willner, K. Garcia-Mansfield, B. Meechooet, R. Reiman, E. Hutchins, M. M. Miller, A. Goel, P. Pirrotte, K. Van Keuren-Jensen, and T. Jovanovic-Talisman, "Single molecule characterization of individual extracellular vesicles from pancreatic cancer," *J. Extracell. Vesicles* **8**(1), 1685634 (2019).
6. K. Bera, K. A. Schalper, D. L. Rimm, V. Velcheti, and A. Madabhushi, "Artificial intelligence in digital pathology - new tools for diagnosis and precision oncology," *Nat. Rev. Clin. Oncol.* **16**(11), 703–715 (2019).
7. J. Chalfoun, M. Majurski, T. Blattner, K. Bhadriraju, W. Keyrouz, P. Bajcsy, and M. Brady, "MIST: accurate and scalable microscopy image stitching tool with stage modeling and error minimization," *Sci. Rep.* **7**(1), 4988 (2017).
8. A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tomancak, V. Hartenstein, and R. J. Douglas, "TrakEM2 Software for Neural Circuit Reconstruction," *PLoS One* **7**(6), e38011 (2012).
9. D. Hörl, F. Rojas Rusak, F. Preusser, P. Tillberg, N. Randel, R. K. Chhetri, A. Cardona, P. J. Keller, H. Harz, H. Leonhardt, M. Treier, and S. Preibisch, "BigStitcher: reconstructing high-resolution image datasets of cleared and expanded samples," *Nat. Methods* **16**(9), 870–874 (2019).
10. X. Y. Lin, B. H. Zhou, and Y. T. Xia, "Online recursive power management strategy based on the reinforcement learning algorithm with cosine similarity and a forgetting factor," *IEEE Trans. Ind. Electron.* **68**(6), 5013–5023 (2021).
11. G. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Psarakis, "A generative model for the joint registration of multiple point sets," in *European Conference on Computer Vision*, (Springer, 2014), 109–122.

12. X. J. Shi, T. Liu, and X. Han, "Improved Iterative Closest Point(ICP) 3D point cloud registration algorithm based on point cloud filtering and adaptive fireworks for coarse registration," *Int. J. Remote Sens.* **41**(8), 3197–3220 (2020).
13. H. Nasir, V. Stankovic, and S. Marshall, "Singular value decomposition based fusion for super-resolution image reconstruction," *Signal Process.-Image Commun.* **27**(2), 180–191 (2012).
14. J. Y. Peng, C. Y. Shi, E. Laugeman, W. G. Hu, Z. Zhang, S. Mutic, and B. Cai, "Implementation of the structural SIMilarity (SSIM) index as a quantitative evaluation tool for dose distribution error detection," *Med. Phys.* **47**(4), 1907–1919 (2020).
15. N. N. Zhang, J. G. Ni, J. Chen, and Z. Li, "Steady-state mean-square error performance analysis of the tensor LMS algorithm," *IEEE Trans. Circuits Syst. II* **68**(3), 1043–1047 (2021).
16. H. Zhang, Z. H. Tang, Y. F. Xie, X. L. Gao, Q. Chen, and W. H. Gui, "A similarity-based burst bubble recognition using weighted normalized cross correlation and chamfer distance," *IEEE Trans. Ind. Inf.* **16**(6), 4077–4089 (2020).
17. P. Thevenaz and M. Unser, "User-friendly semiautomated assembly of accurate image mosaics in microscopy," *Microsc. Res. Tech.* **70**(2), 135–146 (2007).
18. S. Preibisch, S. Saalfeld, and P. Tomancak, "Globally optimal stitching of tiled 3D microscopic image acquisitions," *Bioinformatics* **25**(11), 1463–1465 (2009).
19. <https://github.com/SRMLabHUST>